ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Discrete Algorithms 3 (2005) 362–374

JOURNAL OF DISCRETE ALGORITHMS

www.elsevier.com/locate/jda

An exact and polynomial distance-based algorithm to reconstruct single copy tandem duplication trees

Olivier Elemento [a], Olivier Gascuel [b,*]

[a] *Lewis-Sigler Institute for Integrative Genomics, Princeton University, 08544, Princeton, NJ, USA*

[b] *Equipe Méthodes et Algorithmes pour la Bioinformatique, LIRMM, 161 rue Ada, 34392, Montpellier, France*

Available online 11 September 2004

Abstract

The problem of reconstructing the duplication tree of a set of tandemly repeated sequences which are supposed to have arisen by unequal recombination, was first introduced by Fitch (1977), and has recently received a lot of attention. In this paper, we place ourselves in a distance framework and deal with the restricted problem of reconstructing single copy duplication trees. We describe an exact and polynomial distance based algorithm for solving this problem, the parsimony version of which has previously been shown to be NP-hard (like most evolutionary tree reconstruction problems). This algorithm is based on the minimum evolution principle, and thus involves selecting the shortest tree as being the correct duplication tree. After presenting the underlying mathematical concepts behind the minimum evolution principle, and some of its benefits (such as statistical consistency), we provide a new recurrence formula to estimate the tree length using ordinary least-squares, given a matrix of pairwise distances between the copies. We then show how this formula naturally forms the dynamic programming framework on which our algorithm is based, and provide an implementation in $O(n^3)$ time and $O(n^2)$ space, where n is the number of copies.

Keywords: Evolution reconstructions; Phylogenetic trees; Duplication trees; Tandemly repeated sequences; Gene families; Distance-based methods; Least-squares; Minimum evolution principle; Exact algorithm; Dynamic programming

* Corresponding author.

E-mail addresses: elemento@princeton.edu (O. Elemento), gascuel@lirmm.fr (O. Gascuel).

1570-8667/$ – see front matter
doi:10.1016/j.jda.2004.08.013

1. Introduction

Tandemly repeated DNA sequences consist of two or more adjacent copies of a DNA fragment. They arise from tandem duplication, in which a sequence of DNA (which may itself contain several copies) is replaced by two adjacent and identical versions of itself. After this duplication event, the two copies evolve independently; they generally undergo mutational events, and thus become approximate over time. Unequal recombination during meiosis is widely viewed as the predominant biological mechanism responsible for the production of tandemly repeated sequences [1–6], at least when the basic repeated motif is large (e.g., minisatellites, protein domains, entire genes with their upstream and downstream regulatory sequences). The problem of reconstructing the duplication history of tandemly repeated sequences was pioneered by Fitch in 1977 [3]. However, it has not received much attention until recently, probably due to the lack of available repeated sequence data, and also because there has been no dedicated computer program available to reconstruct duplication histories. With the huge amount of data produced by the various whole genome sequencing projects (human, mouse, putter fish, worm, yeast, etc.), this problem has gained a lot of attention, due to the fact that genomes of higher eukaryotes contain a large proportion of repeated sequences (more than 50% in the human genome [7]). Indeed, accurate methods for reconstructing the duplication history of these tandemly repeated sequences would be important tools for studying the evolution of genomes. They should provide deeper insights into the processes, dynamics and mechanisms of gene duplication, which is one of the main biological events that genomes use for creating genes with new functions [1]. Another reason for this recent gain of attention is that duplication histories appears to be new and interesting combinatorial objects [8,9], and that their inference from sequence data yields difficult computational problems.

Most of the recent studies have been devoted to repeated sequences generated by single copy duplication events [10–13]. Indeed, the mechanism of unequal recombination allows simultaneous duplication of several copies, but there is now evidence [3,5,6,10,11] that single copy duplications are predominant over multiple copies duplications, at least with tandemly repeated genes. For example, one of the most famous tandemly arranged gene families, the Antennapedia (*antp*)-class homeobox genes, have been shown to have arisen through repetitive single copy duplications [14].

The series of duplications that has given rise to tandemly repeated sequences can be represented by way of a “duplication tree”, which we formally describe below. A duplication tree which only contains single copy duplications is simply called a “single copy duplication tree”. Reconstructing optimal single copy duplication trees has been shown to be NP-hard within a parsimony framework [13], and several authors described approximation algorithms. Benson and Dong [10] developed a greedy algorithm for reconstructing single copy duplication trees, based on the parsimony criterion. Using simulations, they showed that their algorithm performs better than approximation algorithms based on minimum ordered spanning trees, which themselves guarantee a performance ratio of 2. More recently, Tang et al. [11] described a dynamic programming algorithm within a parsimony framework for the same problem, which is based on the lifting technique [15] and has proven performance guaranty of ratio 2. Later, Tang et al. [12] and Jaitly et al. [13] independently described polynomial time approximation schemes (PTAS) for the single copy

problem (within the same parsimony framework), also obtained using the lifting technique combined with local optimization and dynamic programming.

In this paper, we place ourselves in a distance framework and present an exact and polynomial $O(n^3)$ time and $O(n^2)$ space algorithm for reconstructing the optimal single copy duplication tree, where n is the number of copies. Our algorithm is based on the minimum evolution principle [16,17], and uses as input the matrix of pairwise evolutionary distances [18], calculated from the set of ordered nucleotide or protein sequences. The minimum evolution principle involves selecting the tree with shortest ordinary least-squares length estimate as being the correct tree. Due to the use of this principle, our reconstruction algorithm is statistically consistent [17,19], as opposed to parsimony methods which were shown inconsistent by Felsenstein [20]. The content of this paper is organized as follows. First we describe the duplication model, i.e., the characteristics of the mathematical objects we aim at reconstructing. Then we describe the minimum evolution framework on which our algorithm is based and provide a novel recurrence formula for estimating the length of any given tree, from a matrix of pairwise distances. Using this formula, we describe a dynamic programming algorithm to solve the single copy duplication tree problem under the minimum evolution principle.

2. Duplication model

Assuming unequal recombination as the sole mechanism responsible in generating the copies, Fitch [3], and more recently Tang et al. [11,12] and Elemento et al. [5,6] independently introduced the following duplication model. A duplication history (Fig. 1(a) and (b)) is a rooted tree with n labelled and ordered leaves denoted as $(1, 2, 3, \ldots, n)$, in which internal branching nodes correspond to duplication events. In a real duplication history, the time intervals between consecutive duplications are completely known, and the internal nodes are ordered from top to bottom according to the moment they occurred in the course of evolution. However, in the absence of molecular clock (which is almost always the case), it is not possible anymore to relate the number of mutational events to elapsed time, and both the order between the duplication events of two different lineages and the root location are impossible to recover from the sequences. In this case, we are only able to infer a *duplication tree* (Fig. 1(c)), i.e., an unrooted tree with ordered leaves, whose topology is compatible with at least one duplication history. Recovering the position of the root can sometimes be achieved through the use of rooting procedures (outgroups, midpoint [18]), and creates *a rooted duplication tree* (Fig. 1(d)).

A duplicated fragment may only contain a single copy, in which case we say that the duplication event is a 1-duplication (or a single copy duplication). It may also contains 2, 3 or k copies, in which case we call the duplication event a 2-, 3- or k-duplication. When a rooted duplication tree only contains 1-duplication events (such as Fig. 1(d)), we call it a rooted single copy duplication tree, and it is analogous to a binary search tree. Consequently, the number of single copy rooted duplication trees is equal to the number of

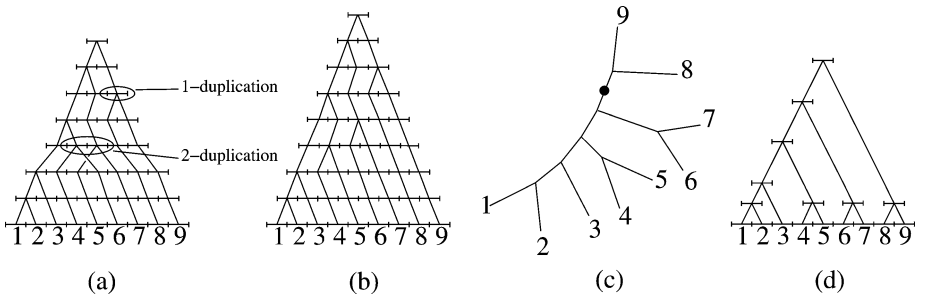

Fig. 1. (a) duplication history; (b) single copy duplication history; (c) single copy duplication tree compatible with history (b); (d) single copy rooted duplication tree obtained when rooting tree (c) on the edge with the bold point.

binary search trees, which is given by the Catalan recursion [21]:

$$C_n = \sum_{k=1}^{n-1} C_k C_{n-k} = \frac{(2n)!}{n!(n+1)!} \sim \frac{4^n}{\sqrt{\pi} n^{3/2}}.$$

As noted in [3] and later in [6], the root of a duplication tree is necessarily located on the path between the most distant copies (i.e., 1 and n) on the locus, simply due to the fact that the root represents the common ancestor of these two copies. In the case of multiple duplications, additional constraints restrict possible root positions [8]. But it is easy to see that a single copy duplication tree can be rooted anywhere along the path between the most distant copies. Suppose that we systematically root single copy duplication trees on the rightmost edge, i.e., the edge associated with n. In this situation, the left subtree is a single copy rooted duplication tree with $n-1$ leaves. Therefore, the number of single copy unrooted duplication trees with n leaves is equal to C_{n-1}. Since this number is exponential (see above), searching for the optimal single copy duplication tree using a trivial algorithm, i.e., one based on exhaustive enumeration of all trees, is impractical when n is large.

A single copy rooted duplication tree $X_{1,n}$, whose leaves are labelled with the ordered set of copies $(1, 2, 3, \ldots, n)$, is obtained by combining two rooted subtrees $X_{1,p}$ and $X_{p+1,n}$ whose leaves are labelled with the ordered sets $(1, 2, 3, \ldots, p)$ and $(p+1, p+2, \ldots, n)$, respectively. Identically, a single copy unrooted duplication tree on $1, 2, \ldots, n$ is obtained by combining two rooted subtrees $X_{1,p}$, $X_{p+1,n-1}$ with elementary subtree $X_{n,n}$ $(1 \leqslant p < n-1)$. In the rest of this paper, the ordered set $(p, p+1, \ldots, q)$ is denoted as $[p, q]$, while, depending on the context, $X_{p,q}$ refers to a rooted tree on $[p, q]$ or to $[p, q]$ itself.

3. Minimum evolution principle and least-squares tree length estimation

3.1. The minimum evolution principle

The minimum evolution (ME) principle [16,17] involves selecting the shortest tree as being the tree which best explains the observed sequences. The tree length is equal to the

sum of edge lengths and edge lengths are estimated by minimizing a least-squares criterion. The problem of inferring optimal phylogenies (i.e., without restriction to duplication trees) within the ME principle is commonly assumed to be NP-hard, as other distance-based phylogeny inference problems [22]. Nonetheless, the ME principle forms the basis of several phylogenetic reconstruction methods, generally based on greedy heuristics. Among them is the popular Neighbor-Joining (NJ) algorithm [23]. Starting from a star tree, NJ iteratively agglomerates external pairs of taxa so as to minimize the tree length at each step. We also recently described FastME, a new software also based on the ME principle but implementing efficient procedures to refine an initial tree by subtree rearrangements, and showed using simulations that it is highly accurate in reconstructing the correct topology [24].

Assuming that we have consistent distance estimators which converge towards the true evolutionary distances as the length of the sequences increases, the ME principle combined with ordinary least-squares (OLS) tree length estimation is statistically consistent [17,19]. Statistical consistency is an essential property in phylogenetic reconstruction, since it ensures that, for the given method and assuming consistent distance estimators (in the case of distance-based methods), the probability of recovering the correct topology increases with sequence length. Inconsistent reconstruction methods, such as parsimony in some cases [20], may converge towards a wrong tree as the amount of data increases. Note that these results were established for any tree topology and then apply to (restricted) duplication trees.

In this section, we introduce a new recurrence formula for estimating the length of any given tree topology using OLS, given a matrix of pairwise evolutionary distances between copies. The application of this general formula to (restricted) single copy duplication trees forms the basis of our reconstruction algorithm.

3.2. Notation

Δ is a matrix of pairwise evolutionary distances between copies, and δ_{ij} is the distance in Δ between copy i and copy j; Υ is an unrooted tree topology, and T represents a valued tree with topology Υ. T induces a matrix of pairwise distances between copies, which we denote Δ^T. In this matrix, δ_{ij}^T denotes the length of the tree path linking copy i and copy j. The sum of the edge lengths of T is denoted as $L(T)$. As shown in Fig. 2, we consider in the rest of this section that T is composed of three non-intersecting subtrees A, B and C. These subtrees are linked together by three edges whose lengths are a, b and c. A is the union of two subtrees A_1 and A_2, and in turn A_1 is the union of two subtrees A_{11} and A_{12}. Two edges with lengths a_1 and a_2 link the root of A to the roots of A_1 and A_2, respectively. In the remainder of this paper, we call R the subset of leaves that do not belong to A (i.e., $R = B \cup C$).

Let X be any subtree of T, and $\overline{X}$ be the average distance in T between the root of X and its leaves. Δ_{XY} and Δ_{XY}^T are the average distances between the leaves of two non-intersecting subtrees X and Y, in the distance matrices Δ and Δ^T, respectively:

$$\Delta_{XY} = \frac{1}{|X||Y|} \sum_{i \in X, j \in Y} \delta_{ij}, \qquad \Delta_{XY}^T = \frac{1}{|X||Y|} \sum_{i \in X, j \in Y} \delta_{ij}^T.$$

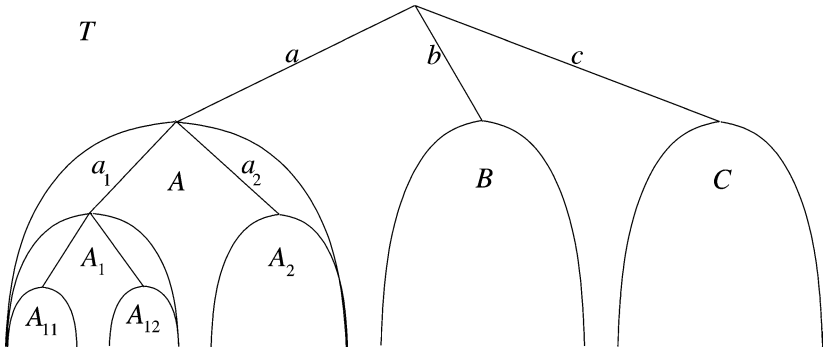

Fig. 2. Unrooted tree T, composed of three subtrees A, B and C.

Given a topology Υ and a distance matrix Δ, the OLS edge length estimation of T is obtained by minimizing the following sum of squares:

$$\sum_{i,j\in T} (\delta_{ij}^T - \delta_{ij})^2.$$

3.3. OLS tree length expression

Theorem. *Let the edges of T be estimated by OLS. Then*:

$$L(T) = \big(L(A) - \overline{A}\big) + \big(L(B) - \overline{B}\big) + \big(L(C) - \overline{C}\big) + \frac{1}{2}(\Delta_{AB} + \Delta_{AC} + \Delta_{BC}). \tag{1}$$

Moreover, $(L(A) - \overline{A})$ is recursively obtained in the following way:
(a) *if A is a leaf, then*

$$\big(L(A) - \overline{A}\big) = 0,$$

(b) *otherwise, $(L(A) - \overline{A})$ is given by*

$$\big(L(A) - \overline{A}\big) = \big(L(A_1) - \overline{A_1}\big) + \big(L(A_2) - \overline{A_2}\big) + \frac{1}{2}\Delta_{A_1A_2} + \frac{1}{2}\left(\frac{|A_2| - |A_1|}{|A|}\right)\Delta_{A_1R} + \frac{1}{2}\left(\frac{|A_1| - |A_2|}{|A|}\right)\Delta_{A_2R}, \tag{2}$$

and the same applies to $(L(B) - \overline{B})$ and $(L(C) - \overline{C})$, by symmetry.

Proof. Using Fig. 2, we see that:

$$L(T) = L(A) + L(B) + L(C) + a + b + c. \tag{3}$$

It has been shown that the average distance between two non-intersecting subtrees X and Y is preserved between Δ and Δ^T, when these subtrees are adjacent to a common ternary node (i.e., A and B, A and C or B and C in Fig. 2), and when edge lengths of T

are estimated by OLS [25–27]. This property holds for any tree topology and then holds for (restricted) duplication trees. Using this property, Δ_{AB}, Δ_{AC} and Δ_{BC} can be expressed in the following way:

$$\begin{aligned}
&\text{(i)} \quad \Delta_{AB} = \Delta^T_{AB} = \overline{A} + a + b + \overline{B},\\
&\text{(ii)} \quad \Delta_{AC} = \Delta^T_{AC} = \overline{A} + a + c + \overline{C},\\
&\text{(iii)} \quad \Delta_{BC} = \Delta^T_{BC} = \overline{B} + b + c + \overline{C},
\end{aligned} \tag{4}$$

and Eq. (1) is obtained by combining Eqs. (3) and (4). Identically, the length of A is equal to the sum of its edge lengths:

$$L(A) = L(A_1) + L(A_2) + a_1 + a_2, \tag{5}$$

while $\overline{A}$ is given by:

$$\overline{A} = \frac{|A_1|}{|A|}(a_1 + \overline{A_1}) + \frac{|A_2|}{|A|}(a_2 + \overline{A_2}). \tag{6}$$

a_1 and a_2 are obtained by rewriting Eq. (4) for A_1, A_2 and R, in place of A, B and C; solving this linear system, we obtain:

$$\begin{aligned}
a_1 &= \frac{1}{2}\Delta_{A_1A_2} + \frac{1}{2}\Delta_{A_1R} - \frac{1}{2}\Delta_{A_2R} - \overline{A_1},\\
a_2 &= \frac{1}{2}\Delta_{A_1A_2} + \frac{1}{2}\Delta_{A_2R} - \frac{1}{2}\Delta_{A_1R} - \overline{A_2}.
\end{aligned} \tag{7}$$

Eq. (2) is finally obtained by subtracting (6) to (5), and replacing a_1 and a_2 by their analytical expression (7), while equality $(L(A) - \overline{A}) = 0$, if A is a leaf, is a direct consequence of the definitions. □

3.4. Properties

In Eq. (1), $(L(A) - \overline{A})$, $(L(B) - \overline{B})$ and $(L(C) - \overline{C})$ only depend on the structure of subtrees A, B and C, respectively. Indeed, Eq. (7) shows that the edge length a_1 depends on the copies in subtrees A_1, A_2 and $R = B \cup C$, but not on the structure of R (i.e., the content of B and C). The same applies with a_2. Identically, this property is valid for edge length a_{11}, which depends on the copies in subtrees A_{11}, A_{12} and $R' = A_2 \cup R$, but not on the structure of R', and therefore not on the structure of R. It can be established in this way that none of the edge lengths in A depends on the structure of R. Therefore, to compute $L(T)$, we independently compute the values for $(L(A) - \overline{A})$, $(L(B) - \overline{B})$ and $(L(C) - \overline{C})$, and then apply Eq. (1).

For the same reasons, $(L(A_1) - \overline{A_1})$ and $(L(A_2) - \overline{A_2})$ only depend on the structure of A_1 and A_2, respectively. Therefore, to compute $(L(A) - \overline{A})$, we independently compute the values for $(L(A_1) - \overline{A_1})$ and $(L(A_2) - \overline{A_2})$, and then apply Eq. (2).

Finally, it has to be noted that the tree length estimate does not depend on the internal node chosen to define the A, B, C partition [25,26], even when this property is not obvious from above theorem.

4. Reconstructing optimal single copy duplication trees under the ME principle

The above recurrence formula enables us to calculate the OLS length of any unrooted tree topology, given a matrix of pairwise distances. In this section, we seek the duplication tree whose length is minimum, among all possible single copy duplication trees. As we shall see, the above formula not only allows tree length estimation, but also forms the basis of a dynamic programming algorithm which solves the problem at hand.

4.1. Basic algorithm

Eq. (1) consists of four independent terms: $(L(A) - \overline{A})$, $(L(B) - \overline{B})$, $(L(C) - \overline{C})$, and the remaining term. As we said above, $(L(A) - \overline{A})$, $(L(B) - \overline{B})$ and $(L(C) - \overline{C})$ only depend on the structure of subtrees A, B and C, respectively, while the remaining term consists of average distances, and therefore does not depend on the structure of A, B and C. To minimize Eq. (1), we adopt a divisive strategy, which consists first in partitioning the whole set of copies into three subsets A, B and C, then in independently computing the structure which minimizes $(L(X) - \overline{X})$ for each of these subsets, and finally in applying Eq. (1). The optimal tree is given by the optimal partitioning. Moreover, the tree length is independent of the node used to define the partitionning (Section 3), and we only need to examine partionnings where one subset, e.g., C, contains a single copy that corresponds to n (Section 2). Identically, to obtain the optimal structure for A, Eq. (2) shows that we need to evaluate every partitioning of A into A_1 and A_2, then to independently compute the structure for A_1 and A_2 which minimizes $(L(X) - \overline{X})$ and finally to select the partitioning which minimizes Eq. (2). The same holds for B by symmetry.

Although used in some divisive clustering methods [28–30], this strategy cannot be used to reconstruct optimal phylogenies when n is large (except for diameter-based optimality criteria [29]), since the number of combinations of subsets is exponential. This is different with single copy duplication trees since we only have to evaluate combinations of two adjacent intervals, and the total number of combinations is $O(n^3)$. A related approach for phylogenetic reconstruction is described by Bryant [31], assuming that splits (taxon subsets) of the inferred tree have to be taken from a known and polynomially sized set of possible splits.

Let S and M be two $(n-1) \times (n-1)$ matrices and $1 \leqslant p < q \leqslant n-1$. $S_{p,q}$ represents the minimal value of $(L(X_{p,q}) - \overline{X_{p,q}})$ where $X_{p,q}$ is any (single copy duplication) subtree with leaves in $[p, q]$, while $M_{p,q}$ represents the position m where $S_{p,q}$ is optimally partitionned (see below). Let $X_{\overline{p,q}}$ represent the subset of copies that do not belong to $X_{p,q}$ (i.e., $X_{\overline{p,q}} = X_{1,p-1} \cup X_{q+1,n}$). Starting from an interval $[1, n]$ representing the n copies and from the distance matrix between these copies, the reconstruction algorithm for single copy duplication trees necessitates the three following steps:

(a) The first step consists in using Eq. (2) to calculate $S_{p,q}$ for a growing interval $X_{p,q}$ of $[1, n-1]$, until $q - p = n - 3$. Computing $S_{p,q}$ requires evaluating the combination of every couple of adjacent intervals $X_{p,m}$ and $X_{m+1,q}$, with m varying from p to $q-1$.

Therefore, using Eq. (2), $S_{p,q}$ is given by:

$$S_{p,q} = \min_{p \leqslant m \leqslant q-1} \begin{bmatrix} S_{p,m} + S_{m+1,q} \\ +\frac{1}{2}\Delta_{X_{p,m}X_{m+1,q}} \\ +\frac{1}{2}\left(\frac{p+q-2m-1}{q-p+1}\right)\Delta_{X_{p,m}X_{\overline{p,q}}} \\ +\frac{1}{2}\left(\frac{2m-p-q+1}{q-p+1}\right)\Delta_{X_{m+1,q}X_{\overline{p,q}}} \end{bmatrix}, \tag{8}$$

while $M_{p,q}$ is the value of m minimizing the above expression. Moreover, we have $S_{p,p} = 0$ for $1 \leqslant p \leqslant n-1$.

(b) The second step consists in using Eq. (1) to search for the intervals $X_{1,m}$, $X_{m+1,n-1}$ which minimizes $L(T)$ when combined with $X_{n,n}$.

(c) In the third step, the complete tree topology is recovered by stepping back through the optimal intervals stored in M. Then, edge lengths are estimated using Eq. (7), starting from pairs of adjacent leaves and moving up until tree root n; average root-to-leaves distances (the $\overline{X}$ terms) are computed using Eq. (6) and used in subsequent applications of Eq. (7).

Algorithm 1. Single copy duplication tree reconstruction algorithm.
input $[1, n]$, the order of the copies, and the distance matrix Δ
output the optimal single copy duplication tree T
$S \leftarrow (n-1) \times (n-1)$ matrix
$M \leftarrow (n-1) \times (n-1)$ matrix
for l from 1 to $n-3$ **do**
  for i from 1 to $n-l-1$ **do**
    compute $S_{i,i+l}$ and $M_{i,i+l}$ using Eq. (8)
  end for
end for
$L^*(T) \leftarrow \infty$
for m from 1 to $n-2$ **do**
  compute $L(T)$ for $X_{1,m}$, $X_{m+1,n-1}$, $X_{n,n}$ using Eq. (1) and S
  if $L(T) < L^*(T)$ **then**
    $L^*(T) \leftarrow L(T)$, $m^* \leftarrow m$
  end if
end for
chose n as root and connect it to X_{1,m^*} and $X_{m^*,n-1}$
create T by recursively divising those subsets using M
estimate edge lengths of T using Eqs. (6) and (7)
return T

This algorithm is summarized above. The number of intervals which need to be evaluated during the first step is $O(n^2)$. Evaluating a single interval using Eq. (8) necessitates the evaluation of $O(n)$ combinations of adjacent sub-intervals. Evaluating a single combination requires the average distances between the $X_{p,m}$, $X_{m+1,q}$ and $X_{\overline{p,q}}$ subsets to be computed, and necessitates $O(n^2)$ time. Therefore, the total time complexity of the first

step is $O(n^5)$. As we show in the next section, the time required to evaluate a single combination of adjacent sub-intervals can be lowered to $O(1)$ using data preprocessing. When using this refinement, the total time complexity of the first step is lowered to $O(n^3)$.

In the second step, we evaluate every pair of intervals $X_{1,m}$, $X_{m+1,n-1}$. Therefore, the number of combinations that need to be tested in the second step is in $O(n)$. As in the previous step, evaluating a single combination requires average distances between intervals to be computed. Therefore, the time complexity of the second step is $O(n^3)$, and can be lowered to $O(n)$ when using preprocessing.

Constructing the tree topology is a simple tree traversal and requires $O(n)$, while edge-length estimation is very close to [26], which is $O(n^2)$, but can here be lowered to $O(n)$ thanks to preprocessing. The total time complexity is $O(n^5)$ in the above "basic" description of our algorithm, and can be lowered to $O(n^3)$ using algorithmic refinements based on data preprocessing. We describe these refinements in the next section.

4.2. *Preprocessing and* $O(1)$ *computation of average distances*

Eqs. (1), (2) (or equivalently (8)) and (7) require the average distances between subsets of copies, which we denote as A, B and C. These subsets define a partition of $[1, n]$, just as in Fig. 2. To calculate these average distances, we use the following lemma.

Lemma 1. *Let A, B, C be any partition of $[1, n]$, and $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$ be the sets of copies that do not belong to A, B and C, respectively (i.e., are the complements of A, B and C); then*:

$$\Delta_{AB} = \frac{1}{2|A||B|}\big(|A|\big(n-|A|\big)\Delta_{A\widetilde{A}} + |B|\big(n-|B|\big)\Delta_{B\widetilde{B}} - |C|\big(n-|C|\big)\Delta_{C\widetilde{C}}\big),$$

and Δ_{AC}, Δ_{BC} are obtained by symmetry.

Proof. Using the average distance definition, we have:

$$|A||B|\Delta_{AB} = \sum_{i\in A, j\in B}\delta_{ij} = \frac{1}{2}\Bigg(\sum_{i\in A, j\in B\cup C}\delta_{ij} + \sum_{i\in B, j\in A\cup C}\delta_{ij} - \sum_{i\in A\cup B, j\in C}\delta_{ij}\Bigg)$$

and the result follows. □

To compute in $O(1)$ any of the average distances that are required in Eqs. (1), (2), (7) and (8), it is then sufficient to know all average distances between any interval $X_{p,q}$ and its complementary set $X_{\overline{p,q}}$. Our preprocessing involves computing these values for all intervals of $[1, n]$. This is achieved using the following lemma.

Lemma 2. *Let $1 \leqslant p < q \leqslant n$; then*:

$$\Delta_{X_{p,q}X_{\overline{p,q}}} = \frac{1}{(q-p+1)(n-q+p-1)}\begin{pmatrix}(q-p)(n-q+p)\Delta_{X_{p,q-1}X_{\overline{p,q-1}}}\\ -U_{p,q}+V_{p,q}\end{pmatrix},$$

with

$$\Delta_{X_{p,p}X_{\overline{p,p}}} = \frac{1}{n-1}\sum_{i\in[1,n],\neq p}\delta_{ip},$$

$$U_{p,q} = U_{p+1,q} + \delta_{pq} \quad \textit{and} \quad U_{p,p} = 0,$$
$$V_{p,q} = V_{p+1,q} - \delta_{pq} \quad \textit{and} \quad V_{q,q} = \sum_{i \in [1,n], \neq q} \delta_{iq}.$$

Proof. Using the average distance definition once again, we have

$$(q-p+1)(n-q+p-1)\Delta_{X_{p,q}X_{\overline{p,q}}} = \sum_{i \in [p,q], j \in [1,p-1] \cup [q+1,n]} \delta_{ij}$$
$$= \sum_{i \in [p,q-1], j \in [1,p-1] \cup [q,n]} \delta_{ij} + \sum_{i \in [p,q-1]} \delta_{iq} - \sum_{j \in [1,p-1] \cup [q+1,n]} \delta_{qj},$$

and the result is obtained by setting:

$$U_{p,q} = \sum_{i \in [p,q-1]} \delta_{iq},$$
$$V_{p,q} = \sum_{j \in [1,p-1] \cup [q+1,n]} \delta_{qj}. \qquad \square$$

Using recursions of Lemma 2, we compute all $\Delta_{X_{p,q}X_{\overline{p,q}}}$ average distances in $O(n^2)$. We first initialize the $\Delta_{X_{p,p}X_{\overline{p,p}}} = V_{p,p}/(n-1)$ terms, each of them requiring $O(n)$ operations; then we compute all other $U_{p,q}$ and $V_{p,q}$ terms, each of them requiring $O(1)$ operations; finally, we compute all remaining $\Delta_{X_{p,q}X_{\overline{p,q}}}$ average distances, each of them again requiring $O(1)$ operations. This preprocessing requires $O(n^2)$ time and space, and allows an $O(n^3)$ time complexity for our algorithm in Section 4.1.

5. Conclusion

In this paper, we present an exact algorithm for reconstructing single copy duplication trees from a matrix of evolutionary distances between tandemly repeated sequences, using the minimum evolution criterion. Our algorithm is based on a novel recurrence formula for ordinary least-squares estimation of tree length. Using preprocessing and a dynamic programming approach, we show that computing the optimal single copy duplication tree only requires $O(n^3)$ time and $O(n^2)$ space. It would be interesting to compare the performance of our algorithm with some other approaches, in terms of topological accuracy. Indeed, heuristic methods such as Neighbor-Joining (NJ) [23] or DTSCORE [32] often do well in practice. A recent duplication history reconstruction approach based on tree rearrangements also appears promising [9]. Moreover, NJ could easily be adapted to single copy duplication tree reconstruction by only agglomerating adjacent pairs of taxa. However, an exact algorithm such as ours has performance guaranty and will then avoid some possible (even rare) shortcomings that would trap heuristic approaches into local minima. A direction for further research would be to extend (if possible) our results to multiple duplications and to other distance criteria, such as weighted least-squares [33,34], balanced minimum evolution principle [35], or to demonstrate the NP-hardness of these tasks.

References

[1] S. Ohno, Evolution by Gene Duplication, Springer Verlag, New York, 1970.
[2] G. Smith, Evolution of repeated DNA sequences by unequal crossover, Science 191 (1976) 528–535.
[3] W. Fitch, Phylogenies constrained by cross-over process as illustrated by human hemoglobins in a thirteen-cycle, eleven amino-acid repeat in human apolipoprotein A-I, Genetics 86 (1977) 623–644.
[4] A. Jeffreys, S. Harris, Processes of gene duplication, Nature 296 (1981) 9–10.
[5] O. Elemento, O. Gascuel, M.-P. Lefranc, Reconstruction de l'histoire de duplication de gènes répétés en tandem, in: Actes des Journées Ouvertes Biologie Informatique Mathématiques, 2001, pp. 9–11.
[6] O. Elemento, O. Gascuel, M.-P. Lefranc, Reconstructing the duplication history of tandemly repeated genes, Molecular Biology and Evolution 19 (2002) 278–288.
[7] E. Lander, et al., Initial sequencing and analysis of the human genome, Nature 409 (2001) 860–921.
[8] O. Gascuel, M. Hendy, A. Jean-Marie, S. McLachlan, The combinatorics of tandem duplication trees, Systematic Biology 52 (2003) 110–118.
[9] L.W.L. Zhang, B. Ma, Y. Xu, Greedy method for inferring tandem duplication history, Bioinformatics 19 (2003) 1497–1504.
[10] G. Benson, L. Dong, Reconstructing the duplication history of a tandem repeat, in: Proceedings of Intelligent Systems in Molecular Biology (ISMB1999), 1999, pp. 44–53.
[11] M. Tang, M. Waterman, S. Yooseph, Zinc finger gene clusters and tandem gene duplication, in: Proceedings of International Conference on Research in Molecular Biology (RECOMB2001), 2001, pp. 297–304.
[12] M. Tang, M. Waterman, S. Yooseph, Zinc finger gene clusters and tandem gene duplication, J. Comput. Biol. 9 (2002) 429–446.
[13] D. Jaitly, P. Kearney, G. Lin, B. Ma, Methods for reconstructing the history of tandem repeats and their application to the human genome, J. Comput. System Sci. 65 (2002) 494–507.
[14] J. Zhang, M. Nei, Evolution of Antennapedia-class homeobox genes, Genetics 142 (1) (1996) 295–303.
[15] L. Wang, D. Gusfield, Improved approximation algorithms for tree alignment, J. Algorithms 25 (1997) 255–273.
[16] K. Kidd, L. Sgaramella-Zonta, Phylogenetic analysis: concepts and methods, Amer. J. Human Genetics 23 (1971) 235–252.
[17] A. Rzhetsky, M. Nei, Theoretical foundation of the minimum-evolution method of phylogenetic inference, Molecular Biology and Evolution 10 (1993) 1073–1095.
[18] D. Swofford, P. Olsen, P. Waddell, D. Hillis, Molecular Systematics, Sinauer Associates, Sunderland, MA, 1996, Chapter Phylogenetic Inference, pp. 407–514.
[19] F. Denis, O. Gascuel, On the consistency of the minimum evolution principle of phylogenetic inference, Discrete Appl. Math. 127 (2003) 63–77.
[20] J. Felsenstein, Cases in which parsimony or compatibility methods will be positively misleading, Systematic Zoology 27 (1978) 401–410.
[21] I. Vardi, Computational Recreations in Mathematica, Addison-Wesley, Redwood City, CA, 1991.
[22] W. Day, Computational complexity of inferring phylogenies from dissimilarity matrices, Bull. Math. Biol. 49 (1987) 461–467.
[23] N. Saitou, M. Nei, The Neighbor-Joining method: a new method for reconstructing phylogenetic trees, Molecular Biology and Evolution 4 (1987) 406–425.
[24] R. Desper, O. Gascuel, Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle, J. Comput. Biol. 9 (2002) 687–706.
[25] W. Vach, Least-squares approximation of additive trees, in: O. Opitz (Ed.), Conceptual and Numerical Analysis of Data, Springer, Heidelberg, 1989, pp. 230–238.
[26] O. Gascuel, Concerning the NJ algorithm and its unweighted version, UNJ, in: Mathematical Hierarchies and Biology, in: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Providence, 1997, pp. 149–170.
[27] D. Bryant, P. Waddell, Rapid evaluation of least-squares and minimum-evolution criteria on phylogenetic trees, Molecular Biology and Evolution 7 (1998) 1346–1359.
[28] J. Barthélemy, A. Guénoche, Trees and Proximity Representations, Wiley and Sons, Chichester, UK, 1991.
[29] A. Guénoche, P. Hansen, B. Jaumard, Efficient algorithms for divisive hierarchical clustering with diameter criterion, J. Classification 8 (1991) 5–30.

[30] E.D. Silva, P. Hansen, B. Jaumard, Average linkage divisive hierarchical clustering, J. Classification, in press.
[31] D. Bryant, Hunting for trees, building trees and comparing trees: theory and method in phylogenetic analysis, Ph.D. Thesis, Dept. Mathematics, University of Canterbury, 1997.
[32] O. Elemento, O. Gascuel, A fast and accurate distance-based algorithm to reconstruct tandem duplication trees, in: Proceedings of European Conference on Computational Biology (ECCB2002), Bioinformatics 18 (2002) 92–99.
[33] W. Fitch, E. Margoliash, Construction of phylogenetic trees, Science 155 (1967) 279–284.
[34] J. Felsenstein, An alternating least squares approach to inferring phylogenies from pairwise distances, Systematic Biology 46 (1) (1997) 101–111.
[35] R. Desper, O. Gascuel, Theoretical foundation of the balanced minimum evolution method of phylogenetic inference and its relationship to weighted least-squares tree fitting, Molecular Biology and Evolution 21 (3) (2004) 587–598.